Formalisation and classification of grammar and template-mediated techniques to model and ontology verbalisation

Zola Mahlaza

Department of Computer Science,
University of Cape Town,
Rondebosch, Cape Town, South Africa
E-mail: zmahlaza@cs.uct.ac.za

C. Maria Keet

Department of Computer Science,
University of Cape Town,
Rondebosch, Cape Town, South Africa
E-mail: mkeet@cs.uct.ac.za

Abstract: Computational tools that translate modeling languages to a restricted natural language can improve end-user involvement in modeling. Templates are a popular approach for such a translation and are often paired with computational grammar rules to support grammatical complexity to obtain better quality sentences. There is no explicit specification of the relations used for the pairing of templates with grammar rules, so it is challenging to compare the latter templates' suitability for less-resourced languages, where grammar reuse is vital in reducing development effort. In order to enable such comparisons, we devise a model of pairing templates and rules, and assess its applicability by considering 54 existing systems for classification, and 16 of them in detail. Our classification shows that most grammar-infused template systems support detachable grammar rules and half of them introduce syntax trees for multilingualism or error checking. Furthermore, out of the 16 considered grammar-infused template systems, most do not currently support any of form of aggregation (63%) or the embedding of verb conjugation rules (81%); hence, if such features would be required, then they would need to be implemented from the ground up.

Keywords: ontology verbalisation; model verbalisation, natural language generation, template classification, grammar-infused templates.

Reference to this paper should be made as follows: Zola Mahlaza and C. Maria Keet (xxxx) 'Formalisation and classification of grammar and template-mediated techniques to model and ontology verbalisation', *International Journal of Metadata, Semantics and Ontologies*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Zola Mahlaza is a PhD candidate in Computer Science at the University of Cape Town in South Africa. He received his MSc. and Bsc. (Hons.) degrees in Computer Science at the same institution. His research interests include natural language processing, ontology engineering and conceptual modelling.

Maria Keet is an Associate Professor with the Department of Computer Science, University of Cape Town, South Africa. Her research interests are in knowledge engineering, including ontology engineering, natural language generation, and ontology-driven conceptual modeling, which has resulted in over 100 publications. She wrote a textbook on ontology engineering. Keet obtained her PhD at the KRDB Research Centre, Free University of Bozen-Bolzano, Italy, in 2008. She has worked at FUB, the University of KwaZulu-Natal, and as engineer in industry.

1 Introduction

Domain experts or end users with little to no experience with modeling languages find it difficult to extract the information contained in conceptual models and ontologies (Rector et al., 2004; Kuhn, 2013). This may limit their role in verifying the correctness or task-based completeness of the captured domain knowledge, therewith negatively affecting the quality of the resulting software artefact, if any. If interaction between domain experts and modelers, knowledge engineers, or ontologists is to be supported, then languages with which the domain experts or end-users are already familiar should be leveraged instead of requiring them to learn modeling languages. In that spirit, a number of tools have been built to present modeling languages using natural language in a controlled manner (Safwat and Davis, 2014). While there is variation in the methods used to generate text from modeling languages, these systems typically make use of templates, which are sentence-like structures that have slots where different values can be inserted. For instance, class subsumption can be presented in English using the template `A ... is a ...` hence when given the axiom *SubClassOf(Tree Plant)* from the African wildlife ontology (Keet, 2020) we obtain the text '*A tree is a plant*'.

Templates can have grammatical errors due to their inability to handle grammatical phenomena such agreement. For instance, when the previous template's input is the axiom *SubClassOf(Impala Animal)* we obtain '*A impala is a animal*' instead of the correct form '*An impala is an animal*'. While some authors solve this through the introduction of phrase constructions like using "a(n)" instead of either "a" or "an" (e.g., Jarrar et al. (2006)), there are others (e.g., van Deemter et al. (2005)) who make use of templates that capture various linguistic phenomena through grammar rules — henceforth called *grammar-infused templates*, a term first introduced in previous work (Mahlaza and Keet, 2019) — since that combination has wider applicability across languages. For instance, while the verbalisation of the object property *arrivedBefore* from a transport-related ontology can be managed with such phrases without sacrificing text readability in languages with a few grammatical genders (e.g., `... arrivé(e) avant ...` in French), that is not possible in languages where verbs have to agree with a large number of noun classes, as can be seen in Niger-Congo B languages[1], popular examples of which are Swahili and isiZulu. These languages have verbs that can have between roughly 81-289 combinations[2] of affixes for polypersonal agreement. Overall, the combination of templates with computational grammar rules (CGRs) is appealing because one can maintain the simplicity of templates and add limited linguistic complexity when either the need or the desire for improved quality arises. In such a setup, the nature of the pairing becomes relevant, especially for less-resourced languages (LRLs) where re-use of the scant CGRs is an imperative. It is then important to understand the possible ways of combining templates and CGRs and how they manifest themselves in the various existing grammar-infused templates.

Insights gained from such an analysis may be valuable when building Natural Language Generation (NLG) systems in general but also for ontology and model verbalisation, especially in multilingual settings and/or when one wants to support portable CGRs. However, at the time of writing, there is no specification of the different kinds of relationships between templates and CGRs; hence, it is not possible to compare the suitability of these grammar-infused templates for LRLs. In this paper, we address the lack of a specification of the relationships that exist between templates and CGRs by devising a model of pairing templates and CGRs that emphasises support for LRLs through two relationships: *attachment* and *embedding*. We show the applicability of this model, and its resulting classification scheme, by classifying existing grammar-infused template systems and obtain a classification that can be used as an aid when choosing an appropriate template type when building a new NLG system. In particular, in our classification of existing systems, we focus on identifying the supported input languages, output natural languages, combinations of the relations from the developed model, the use of aggregation in each tool, and the support for the embedding of rules responsible for conjugating verbs.

We demonstrate the usefulness of the model and its accompanying list of classified systems with a use case in model validation via ontology-based question generation. We have found that most existing systems that use grammar-infused templates do so to 1) to introduce a two-dimensional structure to templates through a syntax tree thus, then enable either multilingualism (e.g., Dannélls (2012)) and/or error checking (e.g., van Deemter et al. (2005)) and support detachable CGRs. Moreover, only a few tools support the embedding of verb conjugation rules, and the choice of formalism used to encode the templates may limit the permissible input language for the verbaliser.

This paper is an extended version of work published in Mahlaza and Keet (2019), in two main ways. First, we have formalised the foundations of the notions of bare/traditional and grammar-infused templates with attachment and embedding, defining precisely how templates and grammars can be combined. Second, we increased the number of tools considered for evaluation from 41 to 54 through having broadened our inclusion criteria, and we extended the features used to compare those systems, thus enabling a more thorough comparison. We also evaluated the classification scheme with a new robust use case.

The remainder of this paper is structured as follows. Section 2 discusses related work. Section 3 presents the developed model and the different kinds of grammar-infused template families. Section 4 introduces our classification of existing grammar-infused templates. Section 5 presents the evaluation of the the classification scheme. Section 6 provides the discussion and Section 7 concludes.

2 Related work

There are a number of ways to differentiate text generation systems (see Reiter (1995); Hovy (1997); Busemann and Horacek (1998); Channarukul (1999); van Deemter et al. (2005); Gatt and Reiter (2009); Kuhn (2013); Gatt and Krahmer (2018)). Some methods focus on the categorisation of the methods used to map some internal representation to natural language text (e.g., Reiter (1995); Hovy (1997); Gatt and Krahmer (2018)), some on the comparison of direct vs. complex generation techniques (e.g., Reiter (1995); Busemann and Horacek (1998); Gatt and Reiter (2009); Kuhn (2013)), others on the linguistic complexity of the realisation method or resulting language (e.g., van Deemter et al. (2005); Kuhn (2013)), and the rest compare the features of various NLG systems (e.g., Channarukul (1999)). We will discuss these approaches in depth in the rest of this section.

In the early stages of the NLG field, systems were differentiated based on their use of formal knowledge representations and having reusable components vs. systems that are tailored to specific applications. This differentiation was motivated by the differences in efficiency, development and maintenance costs, and the potential re-use of systems (Busemann and Horacek, 1998). However, the usefulness of this categorisation has become limited because systems from one of the two categories (i.e., in-depth grammar-based realisers) are not widespread due to their restrictions pertaining to input specifications and the controlled language to be generated (Gatt and Reiter, 2009). Since shallow generation systems are still popular, that has created the need for a refined categorisation that can be used to differentiate between systems that fall into this class. For instance, since Controlled Natural Languages (CNLs) have gained popularity for rendering ontologies and models as natural language in addition to their use for authoring them, the PENS classification scheme was developed to aid in comparing CNLs and improve how they are built (Kuhn, 2013). The scheme focuses on the precision, expressiveness, naturalness, and simplicity of the resulting controlled language. However, the scheme can only be used to compare the generated languages, but not the methods of generation. As a result, such a scheme cannot be used when choosing appropriate grammar-infused templates based on their support of different kinds of relationships between the templates and CGRs.

While some categorisations have been devised to differentiate systems based on the (1) componential re-use, (2) use of ontologies, and (3) features of their generated controlled languages, others use the type of method used to map a system's internal representations to text. For instance, Hovy (1997) identified canned text, template-based, phrase-based, and feature-based systems as the categories for NLG systems. In the Hovy (1997) categorisation, the first two systems are considered rudimentary in that one only needs to map the input to either the same string or template to obtain text. However, template-based systems are not necessarily rudimentary as they can be part of systems that use natural language generation techniques, such as determining what information should be communicated, deciding specific words to be used, etc. (Reiter, 1995; van Deemter et al., 2005). Over the years, Hovy's categorisation has been refined as new classes of systems have emerged. In particular, the contemporary form of Hovy's categorisation features hand-coded templates, hand-coded grammars, statistical methods, and neural-based generators (Gatt and Krahmer, 2018; Narayan and Gardent, 2020) where the template-based systems are non-trivial. Nonetheless, the classification schemes can only identify the broad classes; hence, their usefulness is limited when comparing systems in each of the classes. There are categorisations that have been introduced to differentiate between systems that belong in each of the last three categories (i.e., hand-coded grammars, statistical methods, and neural-based generators); however, none exists for template-based systems.

It is possible to differentiate between existing statistical methods as they all use a small grammar-based generator and a stochastic re-ranker to generate the most likely form of the output sentence; hence, they can be compared based on their chosen grammar formalism and the ranking method (Gatt and Krahmer, 2018). Hand-crafted grammar-based realisers have been differentiated based on various characteristics such as whether they are responsible for planning in addition to surface realisation, whether they are deterministic, and whether they traverse the input structure or the grammar rules when generating text (Channarukul, 1999, p16-20). Neural methods have been differentiated based on the type neural network (e.g., recurrent and convolutional neural networks) and end-to-end vs. pipeline architectures (Narayan and Gardent, 2020). Even though it is known that templates differ, as some have grammar rules (van Deemter et al., 2005), there has been no effort to create a classification scheme based on the relations that exist for pairing templates and CGRs.

There is a need for a categorisation that can be used for template-based methods. Such a categorisation needs to include the relationships that exist between templates and their CGRs, if they have any, because the re-usability of the rules is an imperative when choosing a type of template for LRLs. In the next section, we introduce our model that focuses on the aforementioned relationships and introduce our classification scheme.

3 Pairing templates and grammar rules

In the following sections, we introduce the model, focusing on two relations, and formally define the two relations. We also present the different kinds of grammar-infused template families that exist, which are based on support for the various combinations of the two aforementioned relations (and their sub-relations), and illustrate how a collection of grammar-infused templates can be determined as belonging to each family.

3.1 Relationships

We propose a model that has two kinds of relationships, *attachment* and *embedding*, that exist between traditional templates and CGRs, as depicted in Figure 1. The attachment relation can be differentiated into two based on mode of participation: , *compulsory* and *partial attachment*. In order to formally define these relations, we first define traditional templates and introduce our notation for CGRs. A traditional, or bare, template is define as follows:

Definition (Bare template): *A bare, or traditional, template is a triple $\langle L, f_o, f_s \rangle$, where:*

- $L = W \cup S \cup A$ *is the template lexicon where* S *is the non-empty set of slot names,* W *is the non-empty set of words and meaningful word fragments from the natural language, and* A *is the set of the language's word separators/dividers;*
- $f_o : L \rightarrow L^n$ *is the template's lexicon ordering function whose output has at least one slot and one word; hence, the number of lexical items in the template is* $n \geq 2$*;*
- f_s *is the function such that,* $f_s : W \times S \rightarrow \{0, 1\}$*, i.e., outputs whether a word is permitted to be used in each slot. All words evaluating to true/1 are assigned to a subset* $X \subset W$.

Traditional templates have minimal linguistic rules to which they must adhere. Their rules pertain to the order of words through f_o, and other rules pertain to the semantics of each slot through f_s.

In order to define the relations, we first define the CGRs that can be associated with bare templates in order to form grammar-infused templates, as follows:

Definition (Associable grammar rules): *Let* Γ *be the set of all computational grammar rules of a language, then* Δ *is the set of CGRs associable with bare templates in order to form grammar-infused templates and it meets the following conditions:*

- $\Delta \neq \emptyset$*;*
- $\Delta \subseteq \Gamma \setminus F$ *where* $F = \{f_{o_i} \in T : i \in \mathbb{N}^*\}$*, i.e., is the set of all the language's lexicon ordering functions as used by the bare templates.*

In the above definition, it suffices for our purposes to think of CGRs as functions that operate on words and their features (e.g., morphological features) to obtain words, phrases, and sentences. This definition is sufficiently generic to accommodate all the different extant formalisms that can be used for grammar. For instance, Meaning-Text Theory (MTT) and Grammatical Framework (GF) use different internal structures[3] but can both be viewed as a collection of functions that operate on words and their features despite the differences in their underlying structures.

We then define relations as follows:

Definition (attachment and embedding): *Embedding, attachment, compulsory attachment, and partial attachment are the relations denoted by* embedded*,* attached*,* comp_attached*,* part_attached*, respectively. Given a set of bare templates* T *that uses the associable CGRs* Δ *of size* $k > 0$ *then:*

- *A CGR* $g \in \Delta$ *is embedded to a template* $t \in T$*, denoted* $\mathsf{embedded}(t, g)$*, through the relation* $\mathsf{embedded} = \{(t, g) \in T \times \Delta \quad | \quad t = \emptyset \iff g = id\}$ *where* id *denotes the identity function;*
- *The set of CGRs* $G_1 \subseteq \Delta$ *is attached to the set of templates* $T_1 \subseteq T$*, denoted* $\mathsf{attached}(T_1, G_1)$*, through the relation* $\mathsf{attached} = \{(T_1, G_1) \in 2^T \times 2^\Delta \mid T_1 \setminus T_1 = \emptyset \ \wedge \ |G_1| = k \ \wedge \ \forall g \in G_1 \ (g \neq id)\}$*;*
- *The set of CGRs* $G_1 \subseteq \Delta$ *is compulsory attached to the set of templates* $T_1 \subseteq T$*, denoted* $\mathsf{comp\_attached}(T_1, G_1)$*, through the relation* $\mathsf{comp\_attached} = \{(T_1, G_1) \in 2^T \times 2^\Delta \mid \forall t \in T_1 \ \exists g \in G_1 \ \mathsf{attached}(t, g) \ \wedge \ \forall h \in G_1 \ \exists u \in T_1 \ \mathsf{attached}(u, h)\}$*;*
- *The set of CGRs* $G_1 \subseteq \Delta$ *is partially attached to the set of templates* $T_1 \subseteq T$*, denoted* $\mathsf{part\_attached}(T_1, G_1)$*, through the relation* $\mathsf{part\_attached} = \{(T_1, G_1) \in 2^T \times 2^\Delta \quad | \quad \exists S_1, S_2 \subset T_1 \quad T_1 = S_1 \cup S_2 \quad \wedge \quad \mathsf{comp\_attached} \quad (S_2, G_1) \wedge \neg \mathsf{attached} \ (S_1, G_1)\}$

In the above definition, embedding specifies a existential dependence between a template and a CGR; hence, deletion of the template results in the loss of the grammar rule. When a CGR is lost in this manner, the function that encoded it becomes the identity function and only returns the input it is given as opposed to applying any language specific rule. The embedding relation gives template creators the ability to scaffold traditional templates from within. Attachment specifies an independent existence between a set of templates and a set of CGRs and there are two forms of the relations: partial and compulsory. The relation supports the re-use of CGRs. The two sub-relations of attachment are meant to differentiate between CGRs used by all templates vs. a subset of templates. In particular, a CGR set is compulsorily attached if every template must use at least one rule from it and every rule is used by one template. A CGR set is partially attached to a set of templates if some members of that set do not have any attached CGRs.

We will illustrate all these relations by using the templates from three existing verbalisers designed by Grūzītis et al. (2010); Stevens et al. (2011); Davis et al. (2012). The Latvian verbaliser (Grūzītis et al., 2010) illustrates embedding well: it makes use of CGRs that capture synonymy for some words and these rules are captured using regular expressions; for instance, it uses "(Ikviens|Katrs) ... ir ..." for class subsumption. If those templates were to be removed, then the rules for synonyms would also be removed. Compulsory attachment can be observed in the BeInformed verbaliser (van Grondelle and Gülpers,

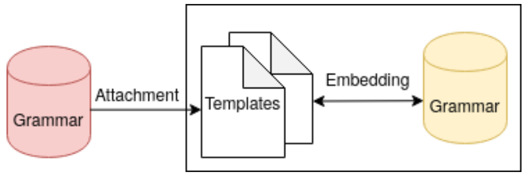

Figure 1 Grammar-infused templates where templates are paired with CGR sets through two kinds of relationships: attachment and embedding. The bidirectional arrow between the template and embedded CGRs denotes their co-dependent existence (embedding) whereas unidirectional arrow denotes independent existence (attachment).

2011; Davis et al., 2012): all its templates require syntax rules from GF's Resource Grammar Library (RGL) that persist even when the templates are deleted. Lastly, partial attachment can be seen in the SWAT verbaliser (Stevens et al., 2011). Most of its templates have attached syntax rules used for obtaining well-formed text in some slots, but there are also templates that do not have such rules.

These relations give rise to different families of grammar-infused templates, and will be introduced in next section.

3.2 Families

Given traditional templates and CGRs, there are seven ways how the templates and CGRs can be combined to obtain grammar-infused templates, which are depicted graphically in Figure 2. These families arise based on a grammar-infused template's support for relations introduced in the previous section. They will be formally defined and illustrated in the remainder of this section.

In order to be able to define the families of grammar-infused templates and illustrate how to classify a collection of grammar-infused templates, we need to first to formally define the concept of grammar-infused template, as follows:

Definition (Grammar-infused template): *A grammar-infused template is a sextuple* $t_g = \langle L, f_o, f_s, A_p, A_c, E \rangle$ *where the first three elements* L, f_o, f_s *are as specified/defined for the bare template and* $A_p \subseteq \beta, A_c \subseteq \Delta, E \subseteq \Delta$ *are associable CGRs. The underlying template* $t = \langle L, f_o, f_s \rangle$ *and associable CGRs must be related in the following manner:*

- embedded(t, g) $\forall g \in E$;
- part_attached$(R,\ A_p)$ *for some* $R \subset T$ *where* $t \in R$ *and* $|R| > 1$;
- comp_attached(R, A_c) *for some* $R \subset T$ *where* $t \in R$ *and* $|R| > 1$.

A set of grammar-infused templates, T_g, can be classified into one of the seven families based on characteristics of the set of embedded CGRs (E), the

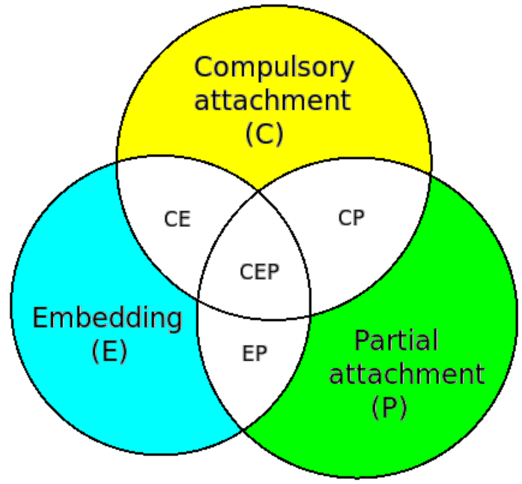

Figure 2 Seven different types of grammar-infused templates. CP, CE, EP, and CEP are combinations of the primary three relations.

set of partially attached CGRs (A_p), and the set of compulsory attached CGRs (A_c). We will illustrate the classification criteria using the four templates in Figure 3 from a hypothetical bilingual verbaliser that generates English and Latvian text. The four templates make use of two types of rules: syntax and synonymy rules. In particular, templates *one* and *two* have syntax rules for constructing singular noun phrases and verb lists using *npList* and *verbList*, respectively (see lines 6 and 11), template *three* is a traditional template hence has no linguistic rules besides word ordering, and template *four* has a rule that encodes synonymy for the adjective *every* (see line 20). The four templates, or any other collection of grammar-infused templates, can be categorised as belonging to one of the seven families if it met the following conditions, respectively:

(1) *P* family: A collection of templates belongs to this family if $A_c = \emptyset$, $E = \emptyset$ and $A_p \neq \emptyset$ (where $\emptyset$ denotes the empty set). For instance, if there were only templates 1-3 in Figure 3, then the only rules that exist in the templates would be the syntax rules. Since the linguistic rules would only be found in two of the three templates, then $A_p \neq \emptyset$, $A_c = \emptyset$, and $E = \emptyset$; hence, the collection would be classified as belonging to this family.
(2) *C* family: A collection of templates belongs to this family if $A_p = \emptyset$, $E = \emptyset$ and $A_c \neq \emptyset$. For instance, if template *one* and *four* were removed in Figure 3, then all the templates would have syntax rules and no other linguistic rules; hence, the collection would be classified as belonging to this family.
(3) *E* family: A collection of templates belong to this family if $A_c = \emptyset$, $A_p = \emptyset$ and $E \neq \emptyset$. For instance, if template *two* and *three* were removed in Figure 3, then only the rules found in template *four* would exist (line 19). Since the only linguistic rule would be embedded, then the collection would be classified as belonging to this family.

```
one(Term, Lexicon) -->
        {Term =.. [sameIndividual|Individuals]},
        {flatten(Individuals, Args)},
        {IndividualList =.. [individualList|Args]},
        [the], [following], [terms], [denote], [the], [same], [individual], [':'],
        npList(and, singular, IndividualList, Lexicon).

two(functionalDataProperty(Properties), Lexicon) -->
        {is_list(Properties), length(Properties, N), N>1},
        [the], [following], [data], [properties], [are], [functional], [':'],
        verbList(singular, Properties, Lexicon).

three(sameIndividual(Individual1,Individual2), Lexicon) -->
        name(Individual1, Lexicon),
        [and],
        name(Individual2, Lexicon),
        [are], [the], [same], [individual].

four(subClassOf(Class1,Class2), Lexicon) -->
        random_select([Ikviens|Katrs], uniform, Every),
        name(Class1, Lexicon),
        [ir],
        name(Class2, Lexicon).
```

Figure 3 Two grammar-infused templates encoded using a Definite Clause Grammar (DCG) in Prolog (Adapted from (Grūzītis et al., 2010; Stevens et al., 2011)

(4) *CP* family: A collection of templates belong to this family if $A_c \neq \emptyset$ and $A_p \neq \emptyset$ and $E = \emptyset$. For instance, if template *three* were removed, template *four*'s synonym rules were redesigned to be attached, and template *four* also had syntax rules attached. In such a scenario, the syntax rules would be compulsorily attached and the synonymy rules would be partially attached; hence, the collection would belong to this family.

(5) *CE* family: A collection of templates belong to this family if $A_c \neq \emptyset$ and $E \neq \emptyset$ and $A_p = \emptyset$. For instance, if template *one* were removed and template *four* also had syntax rules attached, then all templates would syntax rules attached and the only other rules would be embedded; hence, the collection would classified as belonging to this family.

(6) *EP* family: A collection of templates belong to this family if $A_p \neq \emptyset$ and $E \neq \emptyset$ and $A_c = \emptyset$. For instance, we see in Figure 3 that the attached syntax rules are found only in a proper subset of the templates and the synonymy rule is embedded in template *four*, so the collection can be classified as belonging to this family.

(7) *CEP* family: A collection of templates belong to this family if $A_c \neq \emptyset$, $A_p \neq \emptyset$ and $E \neq \emptyset$. For instance, if the verbaliser's templates had additional CGRs that are not synonym or syntax rules that were attached to all templates, then it would belong to this family since it already has partially attached syntax CGRs and the embedded synonym CGR.

We will show the applicability of this scheme by categorising existing systems and highlight the tools' various differences in the next section.

4 Classification of grammar-infused templates

We illustrate the applicability of the categorisation of grammar-infused templates by classifying existing systems. The classification makes it easy to choose an appropriate system from which to bootstrap when building new verbalisers. In particular, it focuses on the tools' input, the type of grammar-infused template used, identifying the family to which the grammar-infused templates belong, the output languages generated by various systems, and determining whether these tools have some form of aggregation or whether generation is achieved by simply slotting in values.

In the previous classification, 41 tools were collected and classified (Mahlaza and Keet, 2019). The previous paper used the following inclusion criteria:

1. Documented verbalisers and NLG systems that utilise grammar-templates.
2. Verbalisers that generated a language other than English.

More specifically, the paper considered seven general linguistic realisers (Gatt and Reiter, 2009), five NLG systems (van Deemter et al., 2005), and 29 systems and verbalisers based on both a recent review (Bouayad-Agha et al., 2014) and a search that specifically focused on finding systems with support for a language other than English. The number of systems considered by the current paper is larger thanks to expanding the inclusion criteria.

We have used a targeted search to expand the list of considered systems. In particular, we searched for recent multilingual verbalisers but not excluding dated systems discovered during the search. This expands the inclusion since the previous criteria was targeted at monolingual systems even though it also unearthed multilingual systems. This yielded 13 additional systems, therewith bringing the total number to 54.

We categorised the systems into one of the five following main NLG approaches, where possible:

(1) *templates*: These are tools that make use of traditional templates only. These templates only have fixed words, slots, and values that can be inserted into the slots. In such systems, multiple templates can be used for variability when generating text from a single input;
(2) *canned text*: Systems that make use of fixed text. Theses systems operate by concatenating short snippets of text together to form sentences;
(3) *grammar*: These tools use hand-crafted fully-fledged grammars for ordering and the inflection of all words — the engine has no fixing of words in a template-like fashion;
(4) *statistical methods*: These tools either make use of grammars that are induced from corpora or use small hand-crafted grammar rules combined with statistical models for re-ranking candidate realisations;
(5) *template + grammar*: These systems make use of grammar-infused templates. This is done either to obtain well-formed text in cases where traditional-templates are insufficient and full-fledged grammar engines are excessive, or to obtain some grammatical variability in the output text;
(6) *other*: Systems that do not fall into any of the previous categories (which may be because insufficient information was presented in the documentation).

The list of all the 54 systems and the categorisation of their realisation approaches are included in Table 1.

The table shows that 13 systems were categorised as using *templates*, 1 *canned text*, 10 *grammar*, 3 *statistical methods*, 16 *templates + grammar*, and 11 were categorised as belonging to the *other* class. In a similar fashion to our previous paper (Mahlaza and Keet, 2019), systems that were not annotated with *template + grammar* were filtered out as out-of-scope, and the remaining 16 systems were then classified into their respective grammar-infused template family using the model introduced in Section 3. Their supported input and other features were also examined. The classification was done manually by analysing the grammar-infused templates found in research papers, technical reports, and PhD theses. The only systems for which source code was also analysed were the ones

Table 1 Literature references to the systems considered and the categorisation of their generation methods

Lit. reference	Generation category
Hewlett et al. (2005)	Templates
Jarrar et al. (2006)	Templates
Ang et al. (2008)	Templates
Al-Yahya (2011)	Templates
Ngonga Ngomo et al. (2019)	Templates
Power and Third (2010)	Templates
Lyudovyk and Weng (2019)	Templates
Liang et al. (2011a)	Templates
Liang et al. (2011b)	Templates
Halpin and Curland (2006)	Templates
Sanby et al. (2016)	Templates
Weal et al. (2007)	Templates
Casteleiro et al. (2011)	Templates
Sadoun et al. (2016)	Canned text
Amith et al. (2017)	Grammar
Hielkema et al. (2007)	Grammar
Aguado et al. (1998)	Grammar
Lavoie and Rainbow (1997)	Grammar
Bateman (1997)	Grammar
Elhadad and Robin (1996)	Grammar
Camilleri et al. (2012)	Grammar
Bouayad-Agha et al. (2012b)	Grammar
Bouayad-Agha et al. (2012a)	Grammar
Dongilli and Franconi (2006)	Grammar
Cimiano et al. (2013)	Statistical methods
Langkilde (2000)	Statistical methods
White (2006)	Statistical methods
van Grondelle and Gülpers (2011); Davis et al. (2008)	Template+grammar
Stevens et al. (2011)	Template+grammar
Androutsopoulos et al. (2013)	Templates+grammar
Dannélls (2012)	Template+grammar
Lim and Halpin (2016)	Template+grammar
Davis et al. (2012)	Template+grammar
Byamugisha et al. (2016)	Template+grammar
Keet et al. (2017)	Template+grammar
Grūzītis et al. (2010)	Template+grammar
McRoy et al. (2000)	Template+grammar
Busemann (1996)	Template+grammar
van Deemter et al. (2005)	Template+grammar
Kaljurand and Fuchs (2007); Kaljurand (2007)	Template+grammar
Wilcock (2001)	Template+grammar
Stenzhorn (2002)	Template+grammar
Hossain et al. (2019)	Template+grammar
Mellish and Pan (2006, 2008)	Other
Power (2009)	Other
Coch (1996)	Other
White and Caldwell (1998)	Other
Piwek (2003)	Other
Cojocaru and Trausan-Matu (2015)	Other
Nguyen et al. (2019)	Other
Halpin and Wijbenga (2010)	Other
Bouttaz et al. (2011)	Other
Bontcheva (2005)	Other
Bontcheva and Wilks (2004)	Other

whose code was publicly released: the SWAT (Stevens et al., 2011), multilingual museum (Dannélls, 2012), NaturalOWL (Androutsopoulos et al., 2013), OWL-ACE (Kaljurand, 2007), Latvian (Grūzītis et al., 2010), and isiZulu (Keet et al., 2017) verbalisers. The classified tools are listed in Table 2.

The table shows that 13 out of the 16 templates do not support the embedding of CGRs to conjugate words. It also shows that 8 of the 16 templates make use of syntax CGRs either for the entire sentence in the case of syntax templates or for a segment of the template in the case of partial syntax templates. Moreover, the various verbalisers and NLG systems expect different kinds of input and 82% of the verbalisers that use grammar-infused templates are designed to generate text from ontologies. All the information in Table 2 can be used to make informed decisions when deciding to bootstrap a new system from existing templates. In particular, it can be used to identify any given language's existing systems and their corresponding grammar-infused templates. It can also be used to determine the suitability of each system's template type based on their supported input type and the characteristics of the realisation approach.

We will illustrate how classification is done by using three verbalisers: Hossain et al. (2019); Keet et al. (2017); Dannélls (2012):

The entity-relationship model verbaliser created by Hossain et al. (2019) currently supports only English as output, its templates are captured through a DCG in Prolog, and the templates have syntax trees that cover only a segment of the template. The templates verbalise entity, relationship, attribute, and constraint declarations by using linguistic rules captured through the functions `vp`, `np` and `verb`. The attached rules are responsible for constructing verb phrases, noun phrases, and conjugating verbs. Since all the templates use those attached rules and there are no other CGRs used, they were classified as belonging to the C family of compulsory attachment.

Keet et al. (2017) built a verbaliser proof-of-concept tool for verbalising OWL ontologies in isiZulu, a grammatically complex South African language. Since traditional templates were deemed inadequate for the language due to its complexity, they designed templates—which they call patterns—and accompanying algorithms for generating text. In effect, the combination of templates and algorithms means that the templates make use of three categories of linguistic rules: morphological agreement, noun pluralisation, and verb conjugation. All the templates make use of the embedded morphological agreement rules. The noun pluralisation and verb conjugation rules are attached to a subset of the templates. This means that the templates only have embedded and partially attached linguistic rules hence the verbaliser belongs to the EP family.

Dannélls (2012) built a verbaliser that generates descriptions of museum artworks. It makes use of GF to capture a painting ontology and the verbalisation templates. GF is able capture both the semantic and linguistic information due to its separation of abstract and concrete syntax modules. The verbaliser's templates use syntax rules from GF's Resource Grammar Library. These rules allow the verbaliser to generate multilingual text. The templates all use the CGRs and the deletion of the templates does not result in the deletion of the rules since the RGL's rules are designed to be detachable and therefore reusable. This means that the verbaliser belongs to the C family, since it only has compulsory attached rules.

In the next section, we will evaluate the classification via a use case.

5 Use case: model validation through question generation

We demonstrate the usefulness of the classification within the context of model validation via question generation in conceptual modelling.

The domain modeling process, as conceptualised by (Bertolino et al., 2011, p.1090), has the following steps from the perspective of a modeling expert:

1. Obtain description of domain from domain expert;
2. Create provisional model;
3. Transform model into a mode that is understood by domain experts;
4. Request domain expert feedback;
5. Incorporate feedback;
6. If there are no conflicts, finalise model; otherwise go to step 3.

In this process, a provisional model can only be finalised after a domain expert approves of the contents of the model. For step 3, one can transform a model into a diagrammatic representation, natural language description or only use sample data when presenting it to domain experts who have no expertise in modelling languages. However, interpreting these layperson-friendly representations may also be time-consuming due to the cognitive load associated with a potentially large legend or glossary for diagrammatic representations or due to a possible difficulty in interpreting of potentially unclear automated textual reports. A structured approach that may be more efficient is the use of natural language generation to produce yes/no questions that probe the contents of model, as advanced by Bertolino et al. (2011). This can be incorporated in the activities already discussed, thereby resulting in the workflow depicted in Figure 4. The generation of yes/no questions can also be seen in computer-assisted educational exercises that make use of ontologies (e.g., Chaudhri et al. (2014)).

The current method of building template-based tools to generate these yes/no questions is to follow a two-step design process: selecting constructs to support and designing templates. In particular, one first decides on which constructs from a modelling language to support so as to limit the scope, depending on the expressiveness

Table 2 Classification and various features of grammar-infused templates for eleven verbalisers, three NLG systems, and two realisers that have support for grammar-infused templates (OWL = Web Ontology Language, XML = Extensible Markup Language, DCG = Definite Clause Grammar, and GF = Grammatical Framework)

System/tool	**Family**	**Input**	**Language(s)**	**Template encoding**	**Template type**	**Embedding verb conjugation rules**	**Aggregation**
Verbalisers							
van Grondelle and Gülpers (2011); Davis et al. (2012)	C	GF	English, Dutch	GF	Syntax templates	No	No
Stevens et al. (2011)	P	OWL	English	DCG in Prolog	Partial syntax template with number	No	Yes
Kaljurand (2007)	C	OWL	English	DCG	Partial syntax template	No	No
Lim and Halpin (2016)	P	-	Malay, Mandarin	C#	Pattern	No	No
Androutsopoulos et al. (2013)	EC	OWL	English, Greek	OWL	Sentence plan	Yes	Yes
Grūzītis et al. (2010)	EC	OWL	Latvian	GF	Syntax templates with synonymy	No	No
Davis et al. (2008)	EP	OWL	English	XML	Template	No	No
Byamugisha et al. (2016)	EP	OWL	Runyankore	Java	Pattern	No	No
Keet et al. (2017)	EP	OWL	IsiZulu	Python	Pattern	No	No
Dannélls (2012)	C	OWL/GF	English, French, Italian, Finnish, Hebrew and Swedish	GF	Syntax templates	No	No
Hossain et al. (2019)	C	RuleML/JSON	English	DCG in Prolog	Partial syntax template	No	No
NLG systems							
Stenzhorn (2002)	EP	XML	English, German, French, Italian, Russian, Bulgarian, Turkish	XML	Morphological template	No	No
van Deemter et al. (2005)	EP	-	English, Dutch, German	-	Syntax template	No	-
Wilcock (2001)	EP	XML	-	Extensible Stylesheet Language	Syntax template	No	Yes
Surface realisers							
McRoy et al. (2000)	E	-	English	Template Specification Language	Template	Yes	Yes
Busemann (1996)	E	Generation Interface Language	-	Template Generation Language	Rule	Yes	Yes

of the language. The second step is the design of the templates. This requires decisions to be made regarding the following:

1. *verbalisation*: The choice of words to use in a template to express each construct, especially for morphologically rich languages (e.g. Grūzītis et al. (2010); Keet and Khumalo (2017b)). This can be achieved by surveying the end-user's preference or it could be decided by the template designer.
2. *bootstrapping*: In the event that verbalisation requires complex grammar-infused templates due to the nature of the intended output language and said language is under-resourced, one has to investigate the possibility of bootstrapping templates from a related language to save time and effort (e.g., (Byamugisha, 2019, p81-107)).
3. *slot design*: A decision also has to be made on the type of slots to be used in the templates. One decides on whether to use semantic variables that are *ad hoc* (e.g., Model-T's template (Puzikov and Gurevych, 2018)), linguistically motivated slots types (e.g., CLaRO's user-friendly templates (Keet et al., 2019)), or slot types from ontologies.

In the above process, our classification of existing grammar-infused templates in Table 2 can be used in choosing an existing form of grammar-infused templates prior to creating templates for another language. In particular, the table can be used, when bootstrapping, to identify the grammar-infused methods that exist for a related natural language and identifying the relationship between template and CGRs for a given kind of grammar-infused templates. When building a verbaliser, it can be used to determine whether it is possible to embed CGRs for a conjugating verbs and whether an existing kind of grammar-infused template already supports aggregation. The only other existing technique that has been used to measure the re-usability of grammar-infused templates is a string-edit based measure (see (Byamugisha, 2019, p81-107)). However,

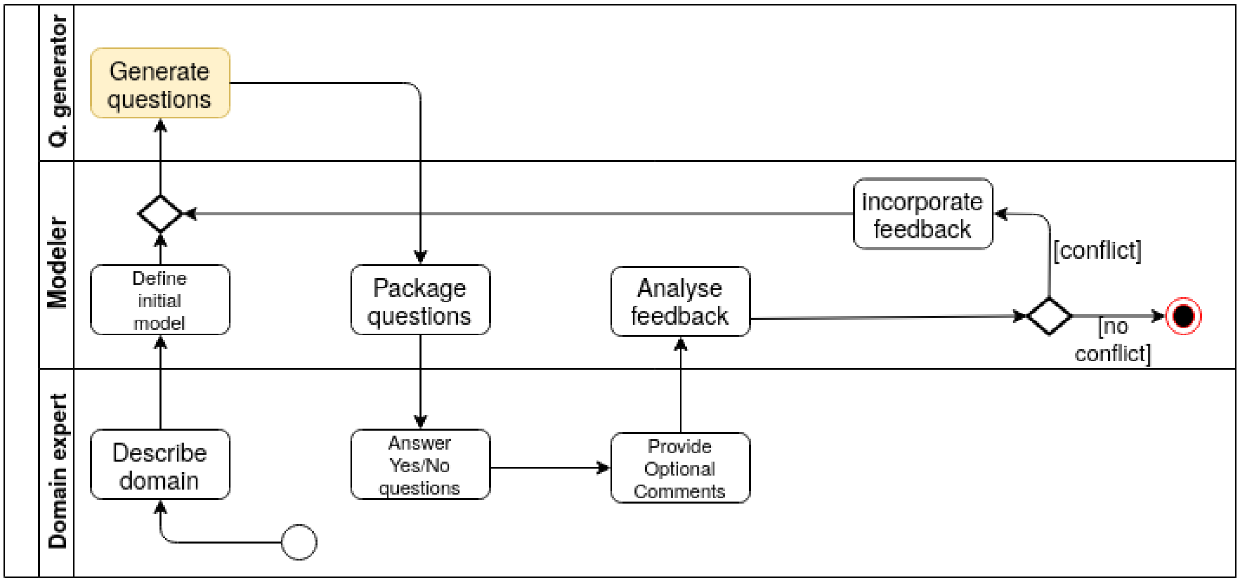

Figure 4 A representation of the activities conducted by the domain expert, modeling expert, and question generator in a domain modeling pipeline where question generation is used by domain experts to probe models. (Adapted from Bertolino et al. (2011))

such a measure can only be applied after a collection of grammar-infused templates has been created for two languages. Unlike our classification, it is unusable as a means of deciding which grammar-infused templates to reuse.

In order to illustrate the benefits of using Table 2, and the classification scheme as a consequence, we consider the generation of yes/no questions in isiZulu for probing the correctness and completeness of a model. We use the African wildlife ontology (Keet, 2020) as an example. There are three OWL constructors that need to be supported in order to interrogate the ontology via a question generator, which are:

1. SubClassOf(C1 C2)
2. ObjectPropertyAssertion(OP I1 I2)
3. DisjointClasses(C1 C2)

For instance, when given the axiom *SubClassOf(Lion Animal)* and the isiZulu lexical items for the concepts *Lion* and *Animal* (*ibhubesi* and *isilwane*, respectively), one can generate the question '*Ingaba azizilwane onke amabhubesi?*' ("Is every lion an animal?"). When deciding to build a system to generate such questions, there are two possibilities:

1) should CGR reuse not be a priority and adequate verbalisation can be achieved using a few templates since there are only three functors, then templates that support the embedding of CGRs (E/CE/EP/CEP families) may suffice as either standalone or as a building block to a larger NLG system.
2) if the reusability of CGRs is a priority and the isiZulu noun pluralisation and verb conjugation rules (Keet and Khumalo, 2017a,b) are adequate, then one can make use of templates that belong to the C/P/CP/CE/EP/CEP families.

Table 2 shows that the only templates that support isiZulu, which have already been shown to be suitable for verbalising class subsumption (Keet et al., 2017), belong to the EP family, so then both possibilities can be supported. However, support for the first possibility would be limited should the other two constructors need to be verbalised via embedding other kinds of CGRs in addition to morphological agreement CGRs.

6 Discussion

The extended classification shows that even though most grammar-infused templates (63%) do not currently support any of form of aggregation, most of them have detachable grammars, and therewith support grammar re-use in some form. It also shows that most grammar-infused templates (81%) do not support the embedding of CGRs to conjugate verbs. This means that additional CGRs cannot be embedded for the purposes of conjugating verbs. When conjugation rules are needed, then they must be attached as opposed to embedded. This limited support for CGR embedding also means that most templates encourage grammar detachability through the attachment relation. Overall, this means that most grammar-infused templates are geared towards LRLs thanks to their support or encouragement of grammar detachability.

There are two main types of templates that use syntax CGRs: syntax templates and partial syntax templates. A syntax template is a template with a syntax tree. A partial syntax template is similar to this, but the syntax tree covers only a subset of the sentence. In both kinds of syntax templates, the syntax CGRs are added to introduce a two-dimensional structure that is either used for multilingual template support or error correction. In particular, the GF-based templates use the syntax tree to obtain multilinguality through the Resource Grammar Library (RGL). This can be seen in the mulitlingual museum verbaliser (Dannélls, 2012) that supports English, French, Italian, Finnish, Hebrew and Swedish through the library. The Dutch soccer report generation system created by van Deemter et al. (2005) uses the syntax trees to check whether referential and quantificational expressions are correct. This shows that that framework used to encode the templates introduces potential differences even when the template types are the same. In particular, while all the mentioned templates attach syntax CGRs, in partial or compulsory manner, the use of GF to encode templates offers multilingual support that is not available in syntax templates that do not use GF.

There are a number of different general-purpose languages (and sometimes markup languages) used to encode the templates and the language used may impact a verbaliser's supported input languages. For instance, in our analysis of the multingual systems (recall Table 2) we see that GF-based tools afford one with multilingual support through the RGL, but they do not generate text by working directly with ontology languages such as OWL. In such tools, the ontologies need to be transferred to GF (e.g., Angelov and Enache (2010)) and this may be time-consuming for large ontologies. This shows that if one intends to introduce syntax CGRs and use GF to support multilingualism and grammar detachability, then that must be weighted against limiting the supported input languages.

7 Conclusion

We have developed a model of pairing templates with natural language CGRs and used it to develop a classification scheme where a particular grammar-infused template formalism may belong to one of seven families. We also showed the applicability of the classification scheme by classifying 16 grammar-infused templates from a pool of 54 verbalisers and NLG systems collected for analysis. We have found that most existing grammar-infused templates are designed to support detachable CGRs as they belong to the partial and compulsory attachment families (C/P/CE/EP families), half of the categorised systems introduce syntax CGRs for the purposes of generating multilingual text and/or ensuring well-formed output text, and only a few tools support the embedding of word conjugation rules. Moreover, the formalism used to encode the templates may limit the language permissible input language. Lastly, most the grammar-infused templates listed in Table 2 do not currently support any of form of aggregation (63%) or the embedding of verb conjugation rules (81%); hence, if such features would be required then they would need to be implemented from the ground up.

Our current work involves building flexible models for grammar-infused templates to support NLG systems that generate sentences in highly agglutinative languages. Since the classification of each collection of templates to its appropriate family is time-consuming, our future work includes the investigation into computational methods to support classification.

References

Aguado, G., Bañón, A., Bateman, J., Bernardos, S., Fernández, M., Gómez-Pérez, A., Nieto, E., Olalla, A., Plaza, R., and Sánchez, A. (1998). ONTOGENERATION: Reusing domain and linguistic ontologies for spanish text generation. In *Proc. of the Workshop on Applications of Ontologies and Problem Solving Methods, associated with the 13th European Conference on Artificial Intelligence (ECAI 1998).* Brighton, United Kingdom, 24-25 August 1998.

Al-Yahya, M. M. (2011). Ontoque: A question generation engine for educational assesment based on domain ontologies. In *ICALT 2011, 11th IEEE International Conference on Advanced Learning Technologies, Athens, Georgia, USA, 6-8 July 2011*, pages 393–395. IEEE Computer Society.

Amith, M., Manion, F. J., Harris, M. R., Zhang, Y., Xu, H., and Tao, C. (2017). Expressing biomedical ontologies in natural language for expert evaluation. In Gundlapalli, A. V., Jaulent, M., and Zhao, D., editors, *MEDINFO 2017: Precision Healthcare through Informatics - Proceedings of the 16th World Congress on Medical and Health Informatics, Hangzhou, China, 21-25 August 2017*, volume 245 of *Studies in Health Technology and Informatics*, pages 838–842. IOS Press.

Androutsopoulos, I., Lampouras, G., and Galanis, D. (2013). Generating natural language descriptions from OWL ontologies: the NaturalOWL system. *Journal of Artificial Intelligence Research*, 48:671–715.

Ang, W. T., Kanagasabai, R., and Baker, C. J. O. (2008). Knowledge translation: Computing the query potential of bio-ontologies. In Burger, A., Paschke, A., Romano, P., and Splendiani, A., editors, *Proceedings of the Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS) Edinburgh, United Kingdom, November 28, 2008*, volume 435 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Angelov, K. and Enache, R. (2010). Typeful ontologies with direct multilingual verbalization. In Rosner, M. and Fuchs, N. E., editors, *Controlled Natural Language - Second International Workshop, CNL 2010, Revised Papers*, volume 7175 of *Lecture Notes in Computer Science*, pages 1–20. Springer. September 13-15, Marettimo Island, Italy.

Bateman, J. A. (1997). Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering*, 3(1):15–55.

Bertolino, A., De Angelis, G., Di Sandro, A., and Sabetta, A. (2011). Is my model right? let me ask the expert. *Journal of Systems and Software*, 84(7):1089 – 1099.

Bontcheva, K. (2005). Generating tailored textual summaries from ontologies. In Gómez-Pérez, A. and Euzenat, J., editors, *The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005, Proceedings*, volume 3532 of *Lecture Notes in Computer Science*, pages 531–545. Springer.

Bontcheva, K. and Wilks, Y. (2004). Automatic report generation from ontologies: The MIAKT approach. In Meziane, F. and Métais, E., editors, *Natural Language Processing and Information Systems, 9th International Conference on Applications of Natural Languages to Information Systems, NLDB 2004, Salford, UK, June 23-25, 2004, Proceedings*, volume 3136 of *Lecture Notes in Computer Science*, pages 324–335. Springer.

Bouayad-Agha, N., Casamayor, G., Mille, S., Rospocher, M., Saggion, H., Serafini, L., and Wanner, L. (2012a). From ontology to NL: generation of multilingual user-oriented environmental reports. In Bouma, G., Ittoo, A., Métais, E., and Wortmann, H., editors, *Natural Language Processing and Information Systems - 17th International Conference on Applications of Natural Language to Information Systems, NLDB 2012, Groningen, The Netherlands, June 26-28, 2012. Proceedings*, volume 7337 of *Lecture Notes in Computer Science*, pages 216–221. Springer.

Bouayad-Agha, N., Casamayor, G., Mille, S., and Wanner, L. (2012b). Perspective-oriented generation of football match summaries: Old tasks, new challenges. *ACM Trans. Speech Lang. Process.*, 9(2):3:1–3:31.

Bouayad-Agha, N., Casamayor, G., and Wanner, L. (2014). Natural language generation in the context of the semantic web. *Semantic Web*, 5(6):493–513.

Bouttaz, T., Pignotti, E., Mellish, C., and Edwards, P. (2011). A policy-based approach to context dependent natural language generation. In Gardent, C. and Striegnitz, K., editors, *ENLG 2011 - Proceedings of the 13th European Workshop on Natural Language Generation, 28-30 September 2011, Nancy, France*, pages 151–157. The Association for Computer Linguistics.

Busemann, S. (1996). Best-first surface realization. In Scott, D., editor, *Proceedings of the Eighth International Natural Language Generation Workshop, INLG 1996*, pages 101–110. June 12-15, Herstmonceux Castle, Sussex, UK.

Busemann, S. and Horacek, H. (1998). A flexible shallow approach to text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation, INLG 1998, Niagara-on-the-Lake, Ontario, Canada, August 5-7, 1998.*

Byamugisha, J. (2019). *Ontology verbalization in agglutinating Bantu languages: a study of Runyankore and its generalizability.* PhD thesis, Department of Computer Science, University of Cape Town, South Africa.

Byamugisha, J., Keet, C. M., and DeRenzi, B. (2016). Bootstrapping a Runyankore CNL from an isiZulu CNL. In Davis, B., Pace, G. J., and Wyner, A. Z., editors, *Proceedings of the 5th International Workshop on Controlled Natural Language, CNL 2016*, volume 9767 of *Lecture Notes in Computer Science*, pages 25–36. Springer. July 25-27, Aberdeen, UK.

Camilleri, J. J., Fuchs, N. E., and Kaljurand, K. (2012). Ace grammar library. Technical Report D11.1., Multilingual Online Translation (MOLTO) project.

Casteleiro, M. A., Des, J., Prieto, M. J. F., Perez, R., and Lekkas, S. (2011). An ontology-based approach to natural language generation from coded data in electronic health records. In Al-Dabass, D., Orsoni, A., Pantelous, A. A., Romero, G., and Félez, J., editors, *Proceedings of the UKSim 5th European Symposium on Computer Modeling and Simulation, EMS 2011, Madrid, Spain, November 16-18, 2011*, pages 366–371. IEEE.

Channarukul, S. (1999). YAG : A template-based natural language generator for real-time systems. Msc thesis, Department of Electrical Engineering and Computer Sciene, University of Wisonsin-Milwaukee, USA.

Chaudhri, V. K., Clark, P. E., Overholtzer, A., and Spaulding, A. (2014). Question generation from a knowledge base. In Janowicz, K., Schlobach, S., Lambrix, P., and Hyvönen, E., editors, *Proceedings of the 19th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2014*, volume 8876 of *Lecture Notes in Computer Science*, pages 54–65. Springer. November 24-28, Linköping, Sweden.

Cimiano, P., Lüker, J., Nagel, D., and Unger, C. (2013). Exploiting ontology lexica for generating natural language texts from RDF data. In Gatt, A. and Saggion, H., editors, *ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013, Sofia, Bulgaria*, pages 10–19. The Association for Computer Linguistics.

Coch, J. (1996). Overview of AlethGen. In Scott et al. (1996), pages 25–28.

Cojocaru, D. A. and Trausan-Matu, S. (2015). Text generation starting from an ontology. In Dardala, M. and Rebedea, T., editors, *12th Romanian Human-Computer Interaction Conference, RoCHI 2015, Bucharest, Romania, September 24-25, 2015*, pages 55–60. Matrix Rom.

Dannélls, D. (2012). *Multilingual text generation from structured formal representations.* PhD thesis, Department of Swedish, University of Gothenburg, Göteborg, Sweden.

Davis, B., Enache, R., van Grondelle, J., and Pretorius, L. (2012). Multilingual verbalisation of modular ontologies using GF and lemon. In Kuhn, T. and Fuchs, N. E., editors, *Proceedings of the Third International Workshop on Controlled Natural Language, CNL 2012*, volume 7427 of *Lecture Notes in Computer Science*, pages 167–184. Springer. August 29-31, Zurich, Switzerland.

Davis, B., Iqbal, A. A., Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., and Handschuh, S. (2008). Roundtrip ontology authoring. In Sheth, A. P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T. W., and Thirunarayan, K., editors, *Proceedings of the 7th International Semantic Web Conference, ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 50–65. Springer. October 26-30, Karlsruhe, Germany.

Dongilli, P. and Franconi, E. (2006). An intelligent query interface with natural language support. In Sutcliffe, G. and Goebel, R., editors, *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006*, pages 658–663. AAAI Press.

Elhadad, M. and Robin, J. (1996). An overview of SURGE: a reusable comprehensive syntactic realization component. In Scott et al. (1996), pages 1–4.

Gatt, A. and Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Gatt, A. and Reiter, E. (2009). SimpleNLG: A realisation engine for practical applications. In Krahmer, E. and Theune, M., editors, *Proceedings of the 12th European Workshop on Natural Language Generation, ENLG 2009*, pages 90–93. The Association for Computer Linguistics. March 30-31, Athens, Greece.

Grūzītis, N., Nespore, G., and Saulite, B. (2010). Verbalizing ontologies in controlled Baltic languages. In Skadina, I. and Vasiljevs, A., editors, *Proceedings of the Fourth International Conference on Human Language Technologies - The Baltic Perspective, Baltic HLT 2010*, volume 219 of *Frontiers in Artificial Intelligence and Applications*, pages 187–194. IOS Press. October 7-8, Riga, Latvia.

Halpin, T. A. and Curland, M. (2006). Automated verbalization for ORM 2. In Meersman, R., Tari, Z., and Herrero, P., editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SeBGIS 2006, Montpellier, France, October 29 - November 3, 2006. Proceedings, Part II*, volume 4278 of *Lecture Notes in Computer Science*, pages 1181–1190. Springer.

Halpin, T. A. and Wijbenga, J. P. (2010). FORML 2. In Bider, I., Halpin, T. A., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., and Ukor, R., editors, *Enterprise, Business-Process and Information Systems Modeling - 11th International Workshop, BPMDS 2010, and 15th International Conference, EMMSAD 2010, held at CAiSE 2010, Hammamet, Tunisia, June 7-8, 2010. Proceedings*, volume 50 of *Lecture Notes in Business Information Processing*, pages 247–260. Springer.

Hewlett, D., Kalyanpur, A., Kolovski, V., and Halaschek-Wiener, C. (2005). Proc. of the workshop on end-user semantic web interaction, held in conjunction with the 4th internatioanl semantic web conference (iswc). In *Proc. of the Workshop on End-User Semantic Web Interaction, held in conjunction with the 4th Internatioanl Semantic Web conference (ISWC)*. CEUR-WS.org. Galway, Ireland, November 7, 2005.

Hielkema, F., Mellish, C., and Edwards, P. (2007). Using WYSIWYM to create an open-ended interface for the semantic grid. In Busemann, S., editor, *Proceedings of the Eleventh European Workshop on Natural Language Generation, ENLG 2007, Schloss Dagstuhl, Germany, June 17-20, 2007*.

Hossain, B. A., Rajan, G., and Schwitter, R. (2019). CNL-ER: A controlled natural language for specifying and verbalising entity relationship models. In Mistica, M., Piccardi, M., and MacKinlay, A., editors, *Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association, ALTA 2019*, pages 126–135. Australasian Language Technology Association. December 4-6, Sydney, Australia.

Hovy, E. (1997). Survey of the state of the art in human language technology. In Cole, R., Mariani, J., Uszkoreit, H., Varile, G. B., Zaenen, A., Zampolli, A., and Zue, V., editors, *The Bantu Languages*, chapter 4, pages 139–146. Cambridge University Press and Giardini.

Jarrar, M., Keet, C. M., and Dongilli, P. (2006). Multilingual verbalization of ORM conceptual models and axiomatized ontologies. Technical report, Starlab, Vrije Universiteit Brussel, Belgium.

Kaljurand, K. (2007). *Attempto Controlled English as a Semantic Web Language*. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu, Estonia.

Kaljurand, K. and Fuchs, N. E. (2007). Verbalizing OWL in Attempto Controlled English. In Golbreich, C., Kalyanpur, A., and Parsia, B., editors, *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Innsbruck, Austria, June 6-7, 2007*, volume 258 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Keet, C. M. (2020). The African wildlife ontology tutorial ontologies. *Journal of Biomedical Semantics*, 11(1):4.

Keet, C. M. and Khumalo, L. (2017a). Grammar rules for the isiZulu complex verb. *Southern African Linguistics and Applied Language Studies*, 35(2):183–200.

Keet, C. M. and Khumalo, L. (2017b). Toward a knowledge-to-text controlled natural language of isiZulu. *Language Resources and Evaluation*, 51(1):131–157.

Keet, C. M., Mahlaza, Z., and Antia, M.-J. (2019). CLaRO: A Controlled Language for Authoring Competency Questions. In *Garoufallou E., Fallucchi F., William De Luca E. (eds) Metadata and Semantic Research. MTSR2019. Communications in Computer and Information Science, vol 1057*, pages 3–15. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-36599-8_1.

Keet, C. M., Xakaza, M., and Khumalo, L. (2017). Verbalising OWL ontologies in isizulu with python. In Blomqvist, E., Hose, K., Paulheim, H., Lawrynowicz, A., Ciravegna, F., and Hartig, O., editors, *The Semantic Web: ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Revised Selected Papers*, volume 10577 of *Lecture Notes in Computer Science*, pages 59–64. Springer. May 28 - June 1, Portorož, Slovenia.

Kuhn, T. (2013). The understandability of OWL statements in controlled English. *Semantic Web*, 4(1):101–115.

Langkilde, I. (2000). Forest-based statistical sentence generation. In *6th Applied Natural Language Processing Conference, ANLP 2000, Seattle, Washington, USA, April 29 - May 4, 2000*, pages 170–177. ACL.

Lavoie, B. and Rainbow, O. (1997). A fast and portable realizer for text generation systems. In *5th Applied Natural Language Processing Conference, ANLP 1997, Marriott Hotel, Washington, USA, March 31 - April 3, 1997*, pages 265–268. ACL.

Liang, F., Stevens, R., and Rector, A. L. (2011a). Ontoverbal-m: a multilingual verbaliser for SNOMED CT. In Montiel-Ponsoda, E., McCrae, J. P., Buitelaar, P., and Cimiano, P., editors, *Proceedings of the 2nd International Workshop on the Multilingual Semantic Web, Bonn, Germany, October 23, 2011*, volume 775 of *CEUR Workshop Proceedings*, pages 13–24. CEUR-WS.org.

Liang, S. F., Stevens, R., Scott, D., and Rector, A. L. (2011b). Automatic verbalisation of SNOMED classes using ontoverbal. In Peleg, M., Lavrac, N., and Combi, C., editors, *Artificial Intelligence in Medicine - 13th Conference on Artificial Intelligence in Medicine, AIME 2011, Bled, Slovenia, July 2-6, 2011. Proceedings*, volume 6747 of *Lecture Notes in Computer Science*, pages 338–342. Springer.

Lim, S. H. and Halpin, T. A. (2016). Automated verbalization of ORM models in Malay and Mandarin. *International Journal of Information System Modeling and Design*, 7(4):1–16.

Lyudovyk, O. and Weng, C. (2019). Snomedtxt: Natural language generation from SNOMED ontology. In Ohno-Machado, L. and Séroussi, B., editors, *MEDINFO 2019: Health and Wellbeing e-Networks for All - Proceedings of the 17th World Congress on Medical and Health Informatics, Lyon, France, 25-30 August 2019*, volume 264 of *Studies in Health Technology and Informatics*, pages 1263–1267. IOS Press.

Mahlaza, Z. and Keet, C. M. (2019). A classification of grammar-infused templates for ontology and model verbalisation. In *Garoufallou E., Fallucchi F., William De Luca E. (eds) Metadata and Semantic Research. MTSR2019. Communications in Computer and Information Science, vol 1057*, pages 64–76. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-36599-8_6.

McRoy, S. W., Channarukul, S., and Ali, S. S. (2000). YAG: A template-based generator for real-time systems. In Elhadad, M., editor, *Proceedings of the First International Natural Language Generation Conference, INLG 2000*, pages 264–267. The Association for Computer Linguistics. June 12-16, Mitzpe Ramon, Israel.

Mellish, C. and Pan, J. Z. (2006). Finding subsumers for natural language presentation. In Parsia, B., Sattler, U., and Toman, D., editors, *Proceedings of the 2006 International Workshop on Description Logics (DL2006), Windermere, Lake District, UK, May 30*

- *June 1, 2006*, volume 189 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Mellish, C. and Pan, J. Z. (2008). Natural language directed inference from ontologies. *Artif. Intell.*, 172(10):1285–1315.

Narayan, S. and Gardent, C. (2020). *Deep Learning Approaches to Text Production*. Morgan & Claypool Publishers.

Ngonga Ngomo, A.-C., Moussallem, D., and Bühmann, L. (2019). A holistic natural language generation framework for the semantic web. *arXiv preprint arXiv:1911.01248*.

Nguyen, V., Son, T. C., and Pontelli, E. (2019). Natural language generation from ontologies. In Alferes, J. J. and Johansson, M., editors, *Practical Aspects of Declarative Languages - 21th International Symposium, PADL 2019, Lisbon, Portugal, January 14-15, 2019, Proceedings*, volume 11372 of *Lecture Notes in Computer Science*, pages 64–81. Springer.

Piwek, P. (2003). A flexible pragmatics-driven language generator for animated agents. In *EACL 2003, 10th Conference of the European Chapter of the Association for Computational Linguistics, April 12-17, 2003, Agro Hotel, Budapest, Hungary*, pages 151–154. The Association for Computer Linguistics.

Power, R. (2009). Towards a generation-based semantic web authoring tool. In Krahmer, E. and Theune, M., editors, *ENLG 2009 - Proceedings of the 12th European Workshop on Natural Language Generation, March 30-31, 2009, Athens, Greece*, pages 9–15. The Association for Computer Linguistics.

Power, R. and Third, A. (2010). Expressing OWL axioms by english sentences: dubious in theory, feasible in practice. In Huang, C. and Jurafsky, D., editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 1006–1013. Chinese Information Processing Society of China.

Puzikov, Y. and Gurevych, I. (2018). E2E NLG challenge: Neural models vs. templates. In Krahmer, E., Gatt, A., and Goudbeek, M., editors, *Proceedings of the 11th International Conference on Natural Language Generation*, pages 463–471. Association for Computational Linguistics. November 5-8, 2018, Tilburg University, The Netherlands.

Rector, A. L., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., and Wroe, C. (2004). OWL pizzas: Practical experience of teaching OWL-DL: common errors & common patterns. In Motta, E., Shadbolt, N., Stutt, A., and Gibbins, N., editors, *Proceedings of the Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, EKAW 2004*, volume 3257 of *Lecture Notes in Computer Science*, pages 63–81. Springer Berlin Heidelberg. October 5-8, 2004, Whittlebury Hall, UK.

Reiter, E. (1995). NLG vs. templates. *CoRR*, cmp-lg/9504013.

Sadoun, D., Mkhitaryan, S., Nouvel, D., and Valette, M. (2016). Readme generation from an OWL ontology describing NLP tools. In Gardent, C. and Gangemi, A., editors, *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web, WebNLG 2016, Edinburgh, UK, September 6, 2016*, pages 46–49. Association for Computational Linguistics.

Safwat, H. and Davis, B. (2014). A brief state of the art of CNLs for ontology authoring. In Davis, B., Kaljurand, K., and Kuhn, T., editors, *Proceedings of the 4th International Workshop on Controlled Natural Language, CNL 2014*, volume 8625 of *Lecture Notes in Computer Science*, pages 190–200, Cham. Springer International Publishing. August 20-22, Galway, Ireland.

Sanby, L., Todd, I., and Keet, C. M. (2016). Comparing the template-based approach to GF: the case of afrikaans. In Gardent, C. and Gangemi, A., editors, *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web, WebNLG 2016, Edinburgh, UK, September 6, 2016*, pages 50–53. Association for Computational Linguistics.

Scott, D., Bateman, J., Lapalme, G., McDonald, D., Paris, C., and Linden, K., editors (1996). *Eighth International Natural Language Generation Workshop (Posters and Demonstrations), INLG 1996, Herstmonceux Castle, Sussex, UK, June 12-15, 1996.*

Stenzhorn, H. (2002). Xtragen - A natural language generation system using XML and Java-technologies. In *The 2nd Workshop on NLP and XML, NLPXML@COLING 2002*. August 24 - September 1, Taipei, Taiwan.

Stevens, R., Malone, J., Williams, S., Power, R., and Third, A. (2011). Automating generation of textual class definitions from OWL to English. *Journal of Biomedical Semantics*, 2(S-2):S5.

van Deemter, K., Theune, M., and Krahmer, E. (2005). Real versus Template-Based Natural Language Generation: A False Opposition? *Computational Linguistics*, 31(1):15–24.

van Grondelle, J. C. and Gülpers, M. (2011). Specifying flexible business processes using pre and post conditions. In Johannesson, P., Krogstie, J., and Opdahl, A. L., editors, *Proceedings of the 4th IFIP WG 8.1 Working Conference on The Practice of Enterprise Modeling, PoEM 2011*, volume 92 of

Lecture Notes in Business Information Processing, pages 38–51. Springer. November 2-3, Oslo, Norway.

Weal, M. J., Alani, H., Kim, S., Lewis, P. H., Millard, D. E., Sinclair, P. A. S., Roure, D. D., and Shadbolt, N. R. (2007). Ontologies as facilitators for repurposing web documents. *Int. J. Hum. Comput. Stud.*, 65(6):537–562.

White, M. (2006). CCG chart realization from disjunctive inputs. In Colineau, N., Paris, C., Wan, S., Dale, R., and Belz, A., editors, *INLG 2006 - Proceedings of the Fourth International Natural Language Generation Conference, July 15-16, 2006, Sydney, Australia*, pages 12–19. The Association for Computer Linguistics.

White, M. and Caldwell, T. (1998). EXEMPLARS: A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation, INLG 1998, Niagara-on-the-Lake, Ontario, Canada, August 5-7, 1998.*

Wilcock, G. (2001). Pipelines, templates and transformations: XML for natural language generation. In *Proceedings of the 1st Workshop on NLP and XML, NLPXML@NLPRS 2001.* November 27-30, Tokyo, Japan.

Notes

[1]This family of languages is sometimes referred to as Bantu languages. However, the phrasing is not used as it has derogatory connotations.

[2]The exact number of combinations for each language varies. It depends on a language's noun classes and the number of semantically permissible subjects and objects for each verb.

[3]MTT is based on dependency-based structures and GF is based on constituency-based structures